國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

Snap2Read/Comp2Watch:
增進手持裝置平台上的多媒體瀏覽體驗

Snap2Read/Comp2Watch: Enhancing the Multimedia Browsing Experience on Mobile Devices

許裕明

Yu-Ming Hsu

指導教授：徐宏民 博士

Advisor: Winston H. Hsu, Ph.D.

中華民國 100 年 7 月

July, 2011

摘　　要

手持裝置服務的興起（例如：電子書、影片串流）以及手持裝置的盛行率不但顯露出在手機上閱讀／觀看的需求,也顯示了在手機服務開發上被看好的商業機會。然而，在不同的服務上都有其所面臨的挑戰。以在手機上閱讀電子書來說，不相容的閱讀器、不一致的電子書格式，甚至是有限的螢幕大小，都會導致在手持裝置上閱讀文件的不便。同時，要在手機上閱讀那些沒有數位副本的紙本雜誌是很困難的。因此，我們提出一個系統「**Snap2Read**」可以自動地切割手機上拍下的文件圖片（從紙本雜誌）並將它們轉成可閱讀的片段（patches）（像是文字、標題、圖片等等）然後將他們縮放、裁切成適合的大小以便讓使用者可以透過手機上的點擊就能夠很簡單地瀏覽數位化的雜誌頁面。另一個在手機上熱門的活動則是觀看影片，但是極小的手機螢幕尺寸、有限的頻寬，以及零碎的使用時間仍然阻礙了使用者的體驗：它們要不就是中斷了使用者的觀看過程，抑或是讓使用者無法一次瀏覽多樣的內容。而傳統的影片摘要技術並不能應用在有限的螢幕上，因此我們提出了「**Comp2Watch**」這個系統，發音近似於「come to watch」。這個名字也有著「將影片畫面組合成美術拼貼」以及「壓縮觀看時間」的意義。它考慮了感興趣的區域（ROI）因素讓使用者能夠快速地瞥過影片，並且我們也修改了價值函數（cost function）用來整合不同長寬比的樣板，我們也處理了因為有限空間而導致的單調排版（monotone layout）問題。實驗結果顯示使用者可以在沒有遺失太多週邊資訊的情況下獲得更清楚的畫面主體。

關鍵字：手持裝置、雜誌、影片、多媒體內容分析、改寫

Abstract

The rise of mobile services (e.g., electronic book, video streaming) and the prevalence of mobile devices reveal the needs for mobile reading/watching and the booming business opportunities in mobile service developments. However, there are different challenges among those services. For reading books on mobile devices, incompatible e-book readers, non-uniformed e-book formats, or even limited screen size causes the inconvenience of reading documents on handheld devices. Meanwhile, it is difficult to read physical magazines that do not have the corresponding digital copies. Therefore, we propose a system, ***Snap2Read***, that can automatically segment the captured document images (i.e., from the physical magazines) in mobile phones into readable patches (e.g. text, title, image), and then scale them into suitable size so that users can easily browse the digitalized magazine pages via the mobile phone with simple clicks. Another popular activity on mobile is watching videos, but the small mobile screen size, low bandwidth, and fragmented watching time also hinder the user experiences: they either interrupt the watching process or limit users to browse many contents at the same time. Traditional video summarization techniques are suffering the small screen issue. Therefore, we propose a system, ***Comp2Watch***, which is pronounced like “come to watch”. It implies the meaning of “composing the frames into a collage” and

“compressing the watching time”. It puts ROI factors into consideration in order to help users take a quick glance at videos. Also, we modify the cost function to incorporate the templates with variable aspect ratios. We also address the monotone layout problem caused by the limited space. The experimental results show that users can obtain clearer subject without losing many contexts.

Keywords: mobile device, magazine, video, multimedia content analysis, adaptation

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

We have proposed two systems to enhance the multimedia browsing experience on mobile devices: ***Snap2Read*** and ***Comp2Watch***. The detail of them are described in Chapter 2 and Chapter 3, respectively.

1.1 Snap2Read

When it comes to reading on handheld device, the presently-available e-book readers (e.g., “Kindle”) and their related software might come to our mind first. Although the release of Apple iPad heated up the competition for e-book readers, the companies that have survived (e.g., Amazon, B&N) stand firm because of their rich resources of digitalized book content. This shows us how important digital content is.

Our work gives another possibility for mobile reading: automatically capture and analyze paper documents from physical magazines that users own and turn them into digitalized pages, adaptively readable in mobile devices (cf. **Figure 1**). Users do not

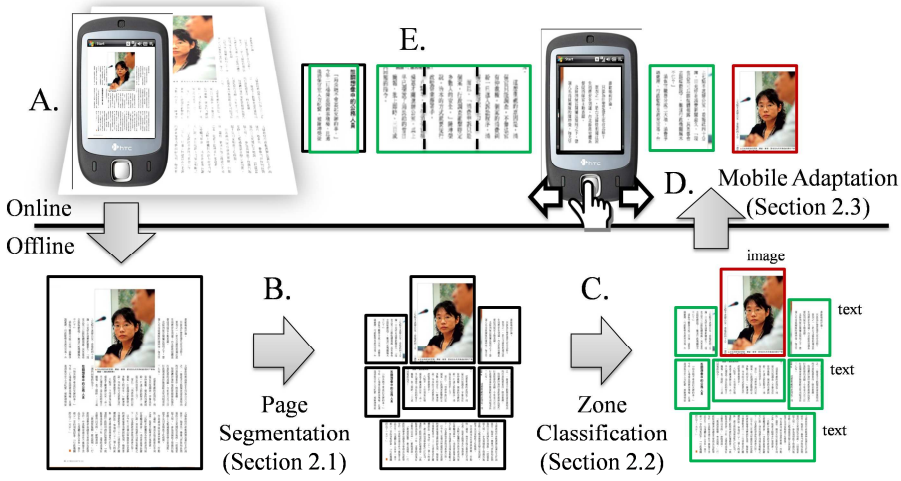


Figure 1. System overview. (A) A user is interested in a physical magazine page and then takes a snapshot at it. (B) Page Segmentation step decomposes the whole page into homogeneous blocks. (C) Zone Classification step predicts the labels on the segmented blocks. (D) Mobile Adaptation step changes the classified blocks into readable patches and further determines their reading sequence. (E) The user can then be guided to read by simple clicks.

have to buy various versions of a certain e-book or their corresponding readers from different companies. As long as it is a paper book (i.e., physical magazine) they own, it can be captured (or scanned), cut into the proper size, and turned into the right format to be read on mobile device. Unlike the traditional process (scanning followed by OCR, i.e., Optical Character Recognition,) which limits reading (i.e., offer text format only, may have error, do not preserve layout, etc.), our approach can preserve the original appearance. Other than panning (i.e., dragging) the whole page while reading, not knowing the present location information and losing track of the reading progress, our system is much more flexible and humanized.

Mobile interaction is a common topic in recent years. Erol et al. [1] tried to link the

physical paper to digital documents. They utilized BWC (Brick Wall Coding) features to retrieve digital documents, and their application aimed at different types of annotation on retrieved document. Liao et al. [2] further provided fine-grained user interface for users, and then the retrieved digital content can be selected, copied or queried. However, they did not address the reading process of users on small-size screen devices.

Putting an entire large resolution document on mobile to read may be considered annoying, but if the content can be rendered properly. Many researches have addressed this issue which focuses on web page browsing. Xie et al. [3] used both spatial and content features to learn a block importance model so that they can classify the "importance" of each segmented block, and then simply aggregate them to the final output list by ranking their importance. Hattori et al. [4] created an "object list" to link each segmented element and proved to be more efficient than commercial mobile web browsers.

The main differences between our work and those works focused on web page reading are as follows. First, segmenting web pages usually takes the "content" into account (i.e., html tags, text information, etc.), while our inputs only have visual features from the snapshots. Second, their output segmented blocks is DOM (Document

Object model) based, which can be further decoded as plain texts. Thus adapting the content to fit the screen has never been a problem, while for document images, the segmented blocks should be further split and padded for different screen resolutions. Third, they only create a simple viewing list for vertical direction reading and do not preserve the layout. By applying the page segmentation techniques on document image, we segment the document directly on the appearance, so the layout of our output does not change.

There are many algorithms can be categorized into several classes in page segmentation researches. (See the reviews in [5].) They usually focus on skew tolerance, time efficiency or certain document types and most of them, unavoidably, need a certain fixed threshold, while our requirement is that the system can be tolerant to different types of layout structure and font size.

The key contributions of *Snap2Read* are:

1. To the best of our knowledge, *Snap2read* represents one of the first attempts that transform physical papers (especially magazines) into electronic readings, thus presenting another venue for mobile reading services.
2. To make reading experience more comfortable, suitably rendering the segmentation blocks on mobile devices is nontrivial. We employ the

classification technique to enhance page adaptation results by knowing the block types.

3. We propose an adaptive morphological approach for page segmentation, which aims to structure captured magazines on mobile devices.
4. Experiments on hundreds of manually collected magazine pages (with segmentation ground truth) show the promising results of our proposed system.

1.2 Comp2Watch

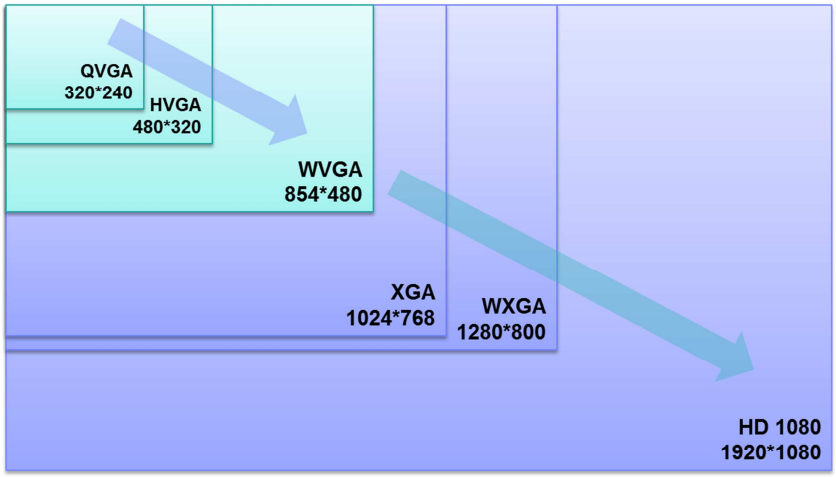
Smart phones have some significant progresses which enabled many things that used to be performed only on the computers. They have already changed the ways of our life. In fact, more and more people are watching videos on mobiles now, and the amount of people who watch videos on mobile devices has been growing rapidly during these past years. The latest report from Nielsen Company [17] shows that the number of Americans watching mobile video has grown more than 40% from 2009 Q4 to 2010 Q4, ending the year at nearly 25 million people. Not only has the popularity grown, the average time that users watching videos on mobile phones has also grown nearly 20% at the same time. At the end of 2010 Q4, people spent 4 hour 20 minutes per month on watching mobile videos in average.

We mainly focus on smart phones instead of pad-like computers because they are

Figure 2. Illustration of baseline (left) and proposed method (right). Applying existing video summarization technique directly on mobile interface which has very limited space may results in some problem. Imagine that putting a huge collage of a video into a smart phone: low zoom level makes the frames unclear, while high zoom level images will occupy lots of spaces.

pocket portable and therefore will be always at hand. However, to the best of our knowledge, there are at least three gaps in mobile video watching:

1. **Fragment watching time:** Users will not watch mobile videos if there are computers or televisions at hand, the most common situation is that when users have only a small chunk of time (e.g., while waiting for a bus, standing in line at a store, or during daily commute in subway). In these situations, users may not have enough time to watch a complete video, and once the watching process is paused, it will not be easy to get back to the time point where users leave last time.



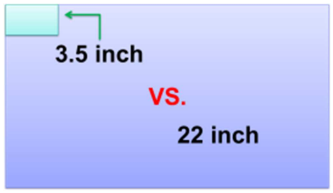


Figure 3. Illustration of screen resolution comparison (top) and screen size comparison (right). Mobile screens are in green color, and LCD screens are in blue color.

2. **Slow/unstable network:** Although online video streaming sites (e.g., Youtube, TED) usually have buffer mechanism to ensure that their videos can be played instantly instead of users' having to download the whole video clip before watching, users cannot get a quick glimpse of the main idea of the content/story until the video ends or has been played for a while. It is an expensive cost in terms of time and also network bandwidth.

3. **The size of mobile display:** In fact, the screen resolution of commercial smart phones (e.g., Sony Ericsson XPERIA X12: 854*480, HTC Desire HD: 800*480) has grown to near the resolution of PC's LCD screens (e.g., XGA standard: 1024*768, WXGA standard: 1280*800). However, when it comes to physical screen size, the smart phones (e.g., Apple iPhone4: 3.5 inch, HTC Desire HD: 4.3 inch) are far behind

from PC's LCD screens (usually more than 20 inch). The comparison above means that smart phones have to put a relatively large content into a tiny space. The detailed illustrations are shown in **Figure 3**.

In addition, it is known that performing actions like pressing virtual buttons on a touch screen is somehow difficult [15]. It is not surprising that interacting with such screens (e.g., selecting, dragging, or clicking) can be very challenging especially when the content is too large so that it must be scaled down.

To handle the first two gaps, we use a collage image composed by selected frames. The time orders of shots are preserved so that it can provide random access via finger click on the touch screen interface on mobiles. It is much more convenient than dragging the timeline beneath the video. Also, downloading a single image instead of the whole video can significantly reduce the network overhead, and taking a glance on the images can enable users to try to get the story in the video or help users to quickly filter out videos which they are not interested in.

The advantages above come from the traditional video summarization techniques. However, the most important issue is the small screen property on such mobile platforms (see **Figure 2**). To bridge this last gap, our intuition is based on the ROI region in the image frame. **Figure 4** shows examples of extracted ROI bounding boxes,

Figure 4. The ROIs in video frames in talk videos (left) and movie trailers (right). White rectangles indicate ROI boundaries.

and we have observed that cropping the ROI regions from the frames has the chance of saving spaces without losing much context information in average cases. The latter user study results support this observation, too.

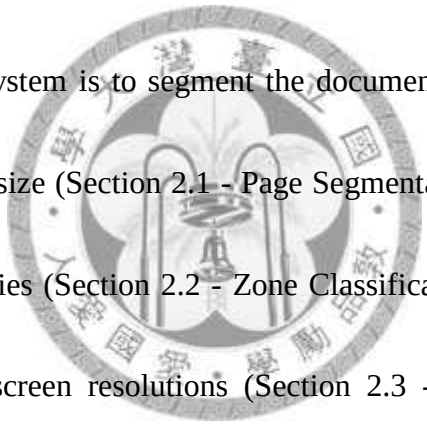
The key contributions of *Comp2Watch* are:

1. To the best of our knowledge, *Comp2Watch* represents one of the first attempts that enables video summarization on mobile devices, thus presenting another experience on mobile video watching.
2. We observed several gaps for mobile video watching and we use ROI extraction to deal with the most challenging one, thus enabling the templates with non-fixed ratio. And the result collage is more compact.
3. We propose several measurements for *Compactness* and evaluate them in the quantitative experiments. For user study, we evaluate both *Clearness* and

Context Loss. These experiments show the promising results of the proposed system.

Chapter 2

Mobile Magazine Reading Enhancement – Snap2Read



The purpose of our system is to segment the document image into homogeneous blocks with the maximum size (Section 2.1 - Page Segmentation), to classify them into a set of predefined categories (Section 2.2 - Zone Classification), and finally to render them to fit for different screen resolutions (Section 2.3 - Mobile Adaptation). The experimental results are shown as well (Section 2.4).

2.1 Page Segmentation

Previous approaches usually focus on one single language or specific types of documents. However, we cannot assume our input to be a certain type of magazines for reading activities on mobile devices, so we do not give any fixed parameters related to character font size, line spacing or layout structure, which traditional page segmentation methods do.

(a) (b) (c) (d)

Figure 5. An illustration of the pre-processing steps: (a) Original image. (b) Binarization. (c) Connected component analysis. Components are in different colors. (d) Noise removal and image block pre-extraction.

Like other morphological methods, our method is a bottom-up approach [5]. The main idea is to group those small connected components into larger regions by dilation. However, we dilate iteratively and automatically select the appropriate dilation kernel.

2.1.1 Pre-Processing

The purpose of the preprocessing step is to filter out noises, dividing lines, and blocks that are possibly images. They tend to be merged with others during dilation, so we must make sure that other main components (mostly texts) will not be affected.

The detailed steps: (1) To take efficiency into consideration, resize the image to a proper area measure (i.e., about 900*650). (2) Do global threshold binarization. (3) For those foreground (i.e., white) pixels, apply connected component analysis. (4) For each component, if its proportion of height to width is significantly high (i.e., 20 times), then remove it from the original image. If the component is big enough (i.e., 1.25%) and its density, the total number of foreground pixels divided by the area size, is high enough

(i.e., 0.1), it will be considered to be “possibly image block” and be extracted in advance. The examples are illustrated in **Figure 5**.

Note that in the step (2), we have also tried edge detector and local threshold binarization method, which are widely used in document image processing [5]. However, although the former depicts the salient edges, it also turns complex image regions into many edge fragments. The latter is mostly used on scanned document images that only contain texts in order to make them robust to lighting change, but it, too, has the similar effect mentioned above on image regions. As a result, we decide to use global threshold binarization to preserve image blocks.

2.1.2 Adaptive Dilation Threshold

In this work, we enlarge small components by dilation operation to group them together, and the square kernel is applied. It enlarges both vertical and horizontal regions of a component, but how to determine the kernel size is vital and nontrivial.

A fact is that the main font size may change from one magazine to another, or from one language to another. Thus, using fixed dilation kernel size for segmentation may not work on every case, so we need to determine it dynamically.

During the iterative dilation process, we find that the number of components will drop rapidly at a certain kernel size, and the most suitable size is at the turning point

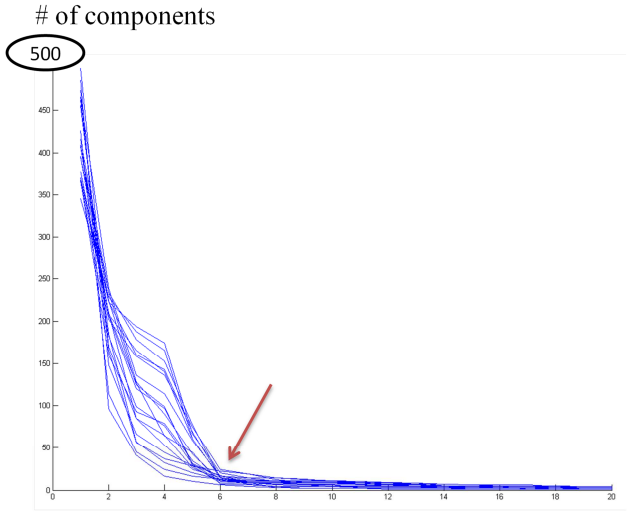
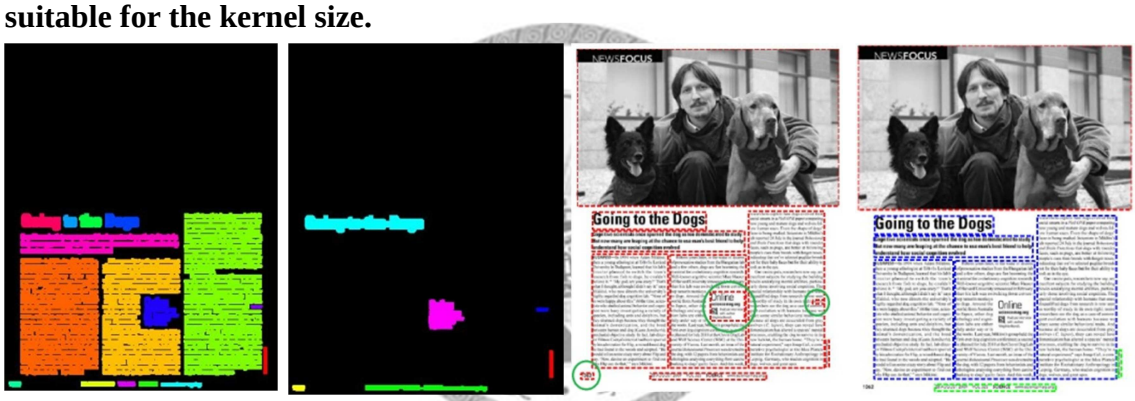


Figure 6. The relationship of component numbers to kernel size, this figure includes 20 pages (blue lines). The red arrow indicates the turning point which is suitable for the kernel size.



(a) (b) (c) (d)

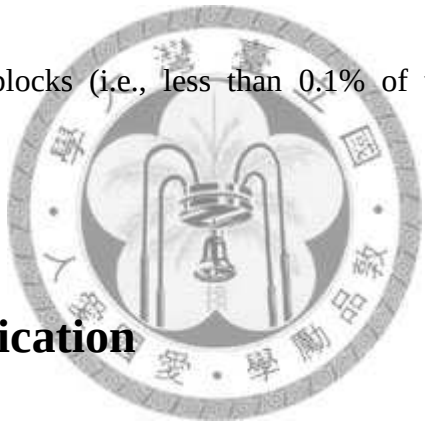
Figure 7. Dilation for little blocks mergence and redundant blocks removal. (a) An example intermediate image from preprocessing step, in which title and footnote are split into several fragments. (b) Remove the large components and dilate again. (c) There exists some inside or non-informative block. (d) The final output example which does not contain those redundant blocks.

(e.g., red arrow in **Figure 6**). The physical meaning of this phenomenon is that a large number of characters and words are merged into lines or paragraphs at the same time.

We then count the number of component versus kernel sizes (e.g. a blue line in **Figure 3**), and the turning point can be found by applying approximate second order differential. To ensure that almost all components are sufficiently merged, we also add a

constraint that the number of components should not be less than a certain number (i.e., 30).

In the last step, minimum enclosing rectangles are extracted from those components which are large enough (i.e., at least 0.3% of whole page area), but some remaining components are not merged well enough (over-segmentation) because the spacing is larger than main texts (e.g., title, footnote, etc.) Therefore, we use a 1.5-time-large kernel to dilate again for the remaining small components (cf. **Figure 7**). Inside blocks and small blocks (i.e., less than 0.1% of whole page area) are also removed (cf. **Figure 7**).



2.2 Zone Classification

The rectangles (blocks) produced by the segmentation step above will be rearranged into meaningful “patches” in accordance with different screen resolutions of the device during the next adaptation step, while images and text blocks have different adaptations: Images cannot be further segmented, but text blocks can be split if they are too large to accommodate themselves to the screen. Therefore, it is necessary to classify images and text blocks (cf. **Figure 1**).

2.2.1 Features for Classification

We use the early fusion scheme – multimodal features concatenated as a long one

– to combine features so that we can learn a classifier by SVM (Support Vector Machine) [6] for each label (e.g., text, image). The three features we used are spatial feature, GCM (Grid Color Moment), and PHOG (Pyramid of Histograms of Orientation Gradients.) [7] Their detailed descriptions are as follows.

Spatial feature contains the coordinate and size of a specified region (i.e., x, y, width and height).

As for **color feature (GCM)**, we adopt the first order moment (mean) and the second order moment (variance) for color feature. The image will be partitioned into several (i.e., 8*8) sub-blocks, and for each block, calculate its mean and variance values. As a result, the GCM feature is a vector with 8 * 8 (blocks) * 3 (color planes) * 2 (moments) = 384 dimensions.

The **shape feature** (**PHOG** descriptor [7]) represents the "local shape," and the "spatial layout" of the image. To calculate PHOG, first extract edge contours by Canny edge detector, and the image is divided into 4^l 44 sub-blocks at level *l*. The HOG of each grid at each pyramid resolution level is then calculated. In this paper, we set level up to 2 (i.e., l = 0, 1, 2) and 8 bins for HOG. Thus, by concatenating different level of resolutions of HOGs, it can be formulated as a vector representation with $(4^0+4^1+4^2)*8=168$ dimensions.

Their dimensions are 4, 384 and 168, respectively. The concatenated feature vector is then measured 556 dimension, and each dimension will be scaled into [-1, +1].

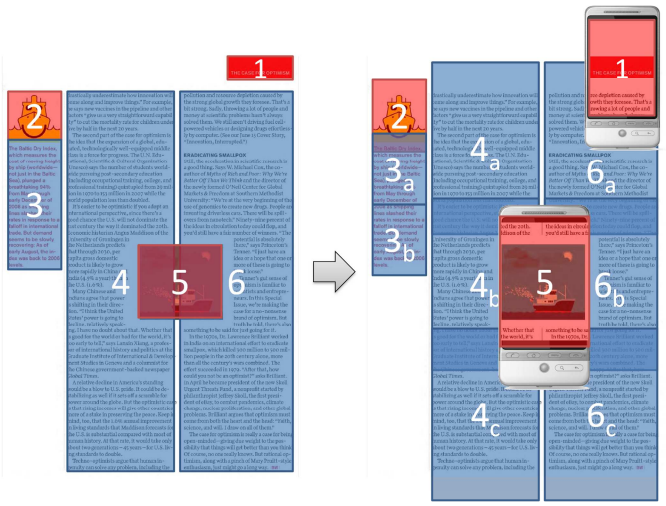
2.2.2 Model Selection

We use the RBF (Radial Basis Function) kernel for classification, so there are two main parameters *g* and *c* to be determined (i.e., gamma and cost, respectively). In order to select the suitable model for prediction, we apply 5-fold cross validation on total 1430 page segments and get average accuracy around 0.95.

2.3 Mobile Adaptation

Although the segmented blocks are composed with homogeneous components (cf. **Figure 9** and **Figure 10**), they cannot be read directly because we extract them as large as possible for each region type, which may not fit the screen resolution, so we must adjust the blocks to readable patches.

As mentioned above, only text blocks may need to be further split (e.g., the wide text block in **Figure 1**(E).). Generally speaking, the English articles tend to stretch in vertical direction, while the Chinese articles tend to expand horizontally. Thus, we adopt a heuristic approach (take English language as an example): For each block, we scale it along its width, and split it into several patches according to its height, and pad those patches whose height are not sufficient to prevent distortion. (cf. **Figure 8**)



(a) (b)

Figure 8. An illustration of adaptation. (a) Original segmented blocks (b) The adapted (scaled, padded) patches. The reading sequence is 1, 2, 3_a, 3_b, 4_a, 4_b, etc…, and it is used to guide users so that they can read the page conveniently through clicks without losing track of page context.

As for the reading sequence, images will have higher priority than texts, and then we rank them from upper left corner to the lower right corner (for Chinese magazines, rank from upper right corner to lower left corner).

We also provide a transparent overview window at the upper right corner of the mobile interface to indicate the current location on whole page. Thus users do not have to zoom-in and zoom-out repeatedly to obtain the geometric information.

2.4 Experimental Results

This section describes our dataset and how we label the ground truth, and it also shows the evaluation of our work.

2.4.1 Magazine Dataset

To the best of our knowledge, there is no public dataset for page segmentation evaluation. Previous page segmentation researches usually tend to use their own private dataset depending either on their target document genre (e.g., newspaper, journal), or on a specific language. Although we know that in recent years, *ICDAR Page Segmentation Competition* has created their own dataset with rich types of sources, only those who participate in the competition can gain access to the dataset. Furthermore, they do not provide documents in Asian languages (e.g. Chinese, Japanese, Korean), which do not have clear bounding boxes for each word, while we expect our system can work well on both type of languages. Thus we create a dataset on our own.

We selected 4 different popular magazines: for Chinese language, "*Common Wealth*" and "*Business Weekly*" are adopted; we also take 2 English magazine named "*Business Week*" and "*Science*". For each magazine, we manually filter out advertisement pages and select 30 scanned pages, and the detail is listed in **Table 1**.

Magazine	Language	# of pages	Scanned resolution
Common Wealth	Chinese	30	1184*1573
Business Weekly	Chinese	30	963*1280
Business Week	English	30	944*1260
Science	English	30	944*1203

Table 1. Magazine dataset for segmentation and classification experiments

To collect groundtruth, we use an editor named GEDI [8], a highly configurable document image annotation tool. It reads an image file, and when annotation is done, it produces a corresponding XML file in GEDI format.

2.4.2 Page Segmentation and Zone Classification Performance

Because the evaluation metrics of the previous methods are usually computed pixel-wise, which is aimed at OCR. However, our output is rectangle-based, which is aimed at locating reading patches. As a result, comparing the two does not make sense.

Although we do not compare them directly, we adopt one of the most widely used metrics in ICDAR 2005 [9], and try to illustrate our performance with their intuition. We have annotated three types of entities (i.e., categories): text, image and footnote. For each entity, the *EDM* (Entity Detection Metric) is calculated.

First, evaluate how much they overlapped between a ground truth zone and a result zone by keeping a global matrix *MatchScore*, which is defined by function

$$MatchScore(i, j) = \begin{cases} \dfrac{T(G_j \cap R_i \cap I)}{T(G_j \cup R_i \cap I)}, if\,(g_j = r_i) \\ 0, otherwise \end{cases} \quad (1)$$

Where I denotes all image pixels, G_j: all pixels inside the ground truth j, R_i: all pixels inside the result i, g_j: the entity type of the ground truth j, r_i: the entity type of the result i, and T(s): a function that counts the elements of set s.

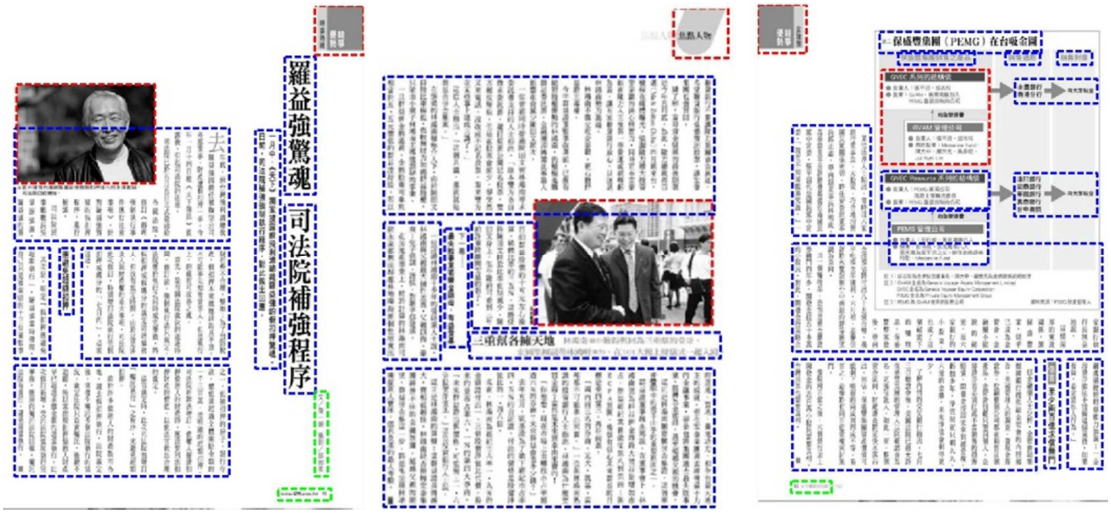
Second, three types of matches are defined (i.e., one-to-one, one-to-many and many-to-one) according to their *MatchScore*: If the *MatchScore* of ground truth zone j and result zone i is higher than the accept threshold (i.e., 0.6), then it is a one-to-one match. (See **Figure 11** for more explanation)

If there are K ground truth zones j_k (k = 1, 2…K) overlapping the same result zone i, and each of their *MatchScore* is between the accept threshold and the reject threshold (i.e., 0.1< $MatchScore(i,j_k)$ < 0.6, k = 1, 2…K), but their summation is higher than the accept threshold, then it is a many-to-one match, and vice versa.

For simplicity, the acceptable matched number for each entity is defined as *MatchNumber* = (w_1*one-to-one + w_2*one-to-many + w_3*many-to-one), where w_1 = 1 and w_2 = w_3 = 0.75 for partial match penalty. Then $DetectRate_t$ and $RecognAccuracy_t$ for entity t are defined as $DetectRate_t$ = $MatchNumber/N_t$, and $RecognAccuracy_t$ = $MatchNumber_t/M_t$. N_t is the number of ground truth regions for t'th entity, and M_t is the number of result regions for t'th entity. The $DetectRate_t$ and $RecognAccuracy_t$ represent

the acceptable ratio among all ground truth zones and all result zones for *t*'th entity, respectively. The Entity Detection Metric score for each entity (text, image, footnote) is then defined as

$$EDM_t = \frac{2 * DetectRate_t * RecognAccuracy_t}{DetectRate_t + RecognAccuracy_t} \quad (2)$$



(a) (b) (c)

Figure 9. Chinese magazine results. (The blue, red and green bounding boxes indicate text, image and footnote, respectively.) (a) A Common Wealth example (b) A Business Weekly example (c) An over-segmentation example which divides a flow chart into text blocks.

(a) (b) (c)

Figure 10. English magazine results. (a) A Business Week example (b) A Science example (c) An over-segmentation example results from figures with unclear bounding boxes.

The overall page segmentation performance is promising. See the breakdowns in **Table 2**. The page segmentation results are sampled in **Figure 9** and **Figure 10**. We also found the results are satisfactory as rendering them in reading patches in two Android phones with different resolutions.

Figure 11. An illustration of accept threshold selection. The groundtruth blocks are marked as magenta and the result blocks are marked as blue. Low MatchScore mostly comes from those small blocks (about 0.8 for logos and 0.6 for footnotes), because a trifling difference (less than 10 pixels) between the two region boundary can result in a large number of percentage of area measure. Thus the smaller blocks tend to have the lower MatchScore. However, this situation does not impede the reading process of users. As a result, we set the accept threshold at 0.6.

The result of Business Weekly seems to have better performance than others (cf. **Table 2**(a).), because its layout is less complicated than others and its text block size is mostly large and rectangle shaped, while Business Week has lower performance on image category (cf. **Table 2**(b).) because it has a lot of figures and tables combined with text explanation inside the bounding boxes, and Footnote category usually has lower performance because parts of them are removed during redudant rectangle removal step, but they are thought of as non-informative blocks, our system does not guide users to read them. Thus the lower performance on footnote category does not really matter.

	Common Wealth			Business Weekly(a)			Business Week			Science		
	T	I	F	T	I	F	T	I	F	T	I	F
N_t	213	44	55	203	49	52	195	91	84	245	77	122
M_t	199	53	42	209	54	40	251	134	67	250	81	79
$DetectRate_t$	0.80	0.93	0.49	0.91	0.91	0.62	0.89	0.76	0.50	0.89	0.81	0.52
$RecognAccuracy_t$	0.86	0.77	0.64	0.89	0.83	0.80	0.70	0.51	0.63	0.88	0.77	0.81
EDM_t	0.83	0.84	0.56	0.90	0.87	0.70	0.78	0.61(b)	0.56	0.88	0.78	0.64

Table 2. Page Segmentation results (T: text. I: image, F: footnote)


國立臺灣大學
教育部獎勵

Chapter 3

Mobile Video Watching Enhancement – Comp2Watch

3.1 Related Work

We have surveyed some kinds of works which are related to our mobile video summarization. Previous works include automatic collage generation, video summarization based on unlimited space, and mobile photo summarization.

Uchihashi, *et al.* [1] was the first work that attempted to propose a comic-like layout summarization on videos, and their key contributions are maintaining time order and enabling the variable frame size in accordance with the importance of a shot, and we use their work as our baseline. Although we do the similar process for video summarization, we not only transplant it to mobile environment, but also take a detailed observation to analyze what has been changed. In section 1, we described three gaps for mobile video summarization, and the first two gaps do not exist on PC environments,

which are strong supports for video summarization on mobile devices; the last gap is the main impedance for such possibility, and we try to settle this problem by introducing ROI extraction.

For collage generation, Rother, et al. [11], Lee, et al. [12] and Goferman, et al. [13] have proposed some of the most representative works. [11] formulates the whole procedure into an energy minimization problem, and they also use graph-cut and Poisson blending to assemble a smooth collage. [12] follows a similar process (i.e., image ranking, ROI selection, ROI packing, and finally image blending) to build a collage. The strength of [12] is that it can be run efficiently on a mobile phone processor. Recently, a work that can compose images with arbitrary ROI into a collage has been proposed [13], the result is more compact and interesting because the space can be filled up with arbitrary shapes, while [11] and [12] only handle rectangle ROIs.

The main difference of our work from them is that their images have no time order like video shots, while our output collage must be time-ordered, and thus this layout problem cannot be solved by their approaches. Most importantly, they do not take the "smallness issue" (the third gap mentioned above) into consideration since a high-level view of the whole collage is enough for their application.

3.2 Keyframe Selection

The main difference from [1] in this step is that we put ROI regions into consideration instead of presenting the whole image. Extracting ROI region can not only enable the flexibility on frame aspect ratio but also benefit the compactness on the whole composed image.

First, we apply shot boundary detection on the given videos and choose the middle frame for each shot as the image presentation of the corresponding shot. We then group these shot images by common hierarchical clustering method, using predefined distance threshold (Section 3.2.1). The importance of each shot will be computed in accordance with shot length, cluster size and ROI ratio to the whole image. Then the importance scores are quantized into certain level to represent the desired template sizes. Finally, we filter out shots that are less important or some shots that are similar within a short period (Section 3.2.2).

3.2.1 Shot Detection and Hierarchical Clustering

For each video, the color histograms of full frames will be extracted for shot detection. We take a common adaptive threshold method: if two adjacent frames or a period of frames are measured to be very different, that will be a shot boundary.

After shot boundaries are detected, we take their middle frame to represent the

corresponding shot and use them as a basic unit in the following steps. For simplicity, we refer to “shot images” as “shots” from now on.

Then a hierarchical clustering step is conducted. The idea of hierarchical clustering is to merge the two closet clusters iteratively. Here we use both color and PHOG [16] features to ensure that the grouped shots are similar not only in terms of color histogram, but also in edge distribution (i.e., shape).

ROI is further detected for each shot using Harel’s work [14]. The ROI region will be cropped and adapted as the final collage representation. What’s more, ROI information plays an important role both on shot importance re-weighting and on layout optimization phase.

3.2.2 Importance Computation

To utilize the space of output collage, the size of all shots must be differentiated by certain criteria. [1] has defined “importance” as “A shot is important if it is both long and rare.” Thus they formulate the importance as the length of a shot normalized by its cluster size to penalize the repeated but discontinuous near-duplicate shots. Therefore the importance of a shot j belongs to cluster k is given by:

$$I_j = L_j \log \frac{1}{W_k} \tag{3}$$

Where $L_j$is the length of the shot j, and W_k (the proportion of shots from the video

that are in cluster k) can be computed from previous clustering result by:

$$W_i = \frac{S_i}{\sum_{j=1}^{C} S_j} \tag{4}$$

S_i is the total length of all shots in cluster i, and C is the total cluster number.

However, we think the importance score should not only reflect the shot length and uniqueness, but also consider ROI propotion on the whole shot; that is, if a shot has a larger ROI region, it should be given a larger template to represent itself (i.e., higher importance score). Therefore we replace the importance by:

$$I_j^{ROI} = I_j \frac{Area_j^{ROI}}{Area_j} \tag{5}$$

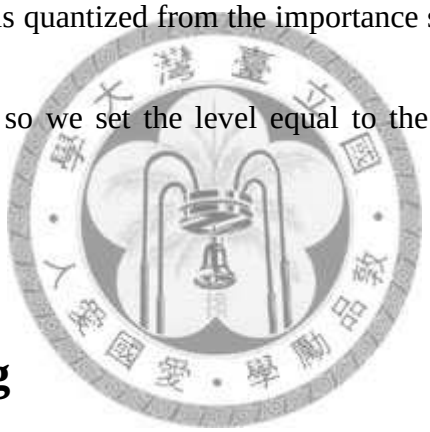
Where $Area_j^{ROI}$ and $Area_j$ are the pixels of the ROI area of shot j and the whole pixels of shot j, respectively.

These importance score will be divided into certain levels during a rough quantization step in order to fit in the pre-defined templates (see **Figure 12**). During this step, some shots will be filtered out (i.e. set their level to zero) if they are not important enough, and others will be assigned sequentially, see **Table 3**.

Table 3. The quantization step from importance score to corresponding level and template size, I is the importance score of a shot (i.e. I^{ROI}), Max is the average of highest ϕ importance score of whole video, here we set ϕ = 5.

Importance Score	Importance Level	Desires Template Size
$I < 1/8\ Max$	0	N/A
$1/8\ Max < I < 2/8\ Max$	1	1*1
$2/8\ Max < I < 3/8\ Max$	2	1*2
$3/8\ Max < I < 4/8\ Max$	4	2*2
$4/8\ Max < I < 6/8\ Max$	6	2*3 or 3*2
$6/8\ Max < I$	9	3*3

The importance level is quantized from the importance score, and it will be used in one of our cost functions, so we set the level equal to the size of its area of desired template.



3.3 ROI Packing

The goal of layout packing algorithm is to put all shots into the given two dimensional space with corresponding size (i.e. importance) while preserving their time order. To achieve this goal, one heuristic way is to arrange those shots into a multi-layered layout (i.e. the whole space is divided into row blocks, and these row blocks contain sub-templates arranged column by column).

Unlike many well studied problems (e.g., bin-packing), such a layout optimization problem that has the above constraints is NP-hard. In order to make the solution feasible,

[1] proposed a “row-block-exhaustive” approach (i.e. optimize each row block one by one). The algorithm is listed as follows:

1. Set the current row block to the top row and the starting shot s = 1.
2. Generate all possible combinations of templates $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots\}$ for current row block.
3. Compute the cost of all combinations and find a combination $\boldsymbol{q}_l$ that has the lowest cost by:

$$l = \arg\min_i \left(\frac{1}{n_i} \sum_{j=1}^{n_i} c\left(f_{s+j-1}, q_{ij}\right) + w_i \right) \quad (6)$$

   Where n_i is the number of shot in combination $\boldsymbol{q}_i$, f_i is the *i*’th shot frame, q_{ij} is the *j*’th template in sequence combination $\boldsymbol{q}_i$, w_i is the remaining space in current row, and $c(x, y)$ is the cost function that measure the difference between the target shot frame image and the matched template.
4. Apply it to current row block and move to the next row block. s is also increased by the length of the solution.
5. Repeat 2. until all frames are packed.

For more detailed information, please refer to [1].

The following three subsections describe the key changes we have made in this algorithm to guarantee that it can work well with the extracted ROIs even in an environment that has a limited space:

We enable the templates with non-fixed aspect ratios since the ROI region is extracted (3.3.1). The cost function has been modified so that it considers not only the importance of a shot, but also its aspect ratio (3.3.2). Inter-row optimization has been introduced to eliminate the monotone layout combinations (3.3.3).

3.3.1 Non-fixed Aspect Ratio Templates

Unlike the baseline approach, we try to enable more flexible templates instead of

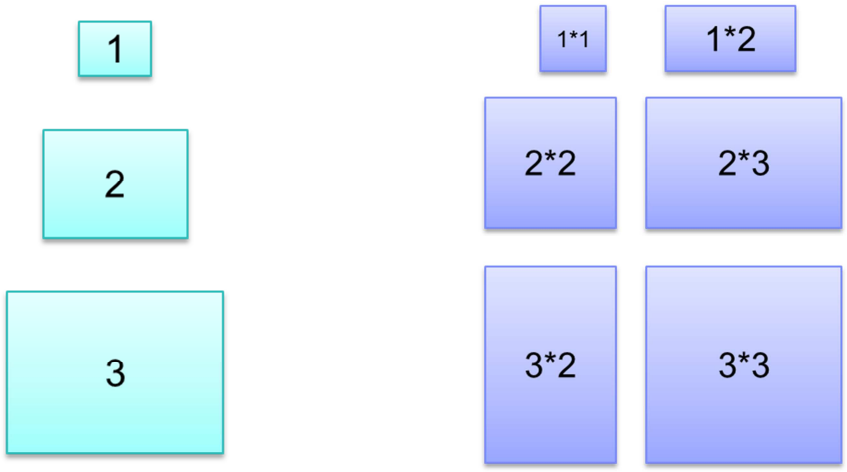


Figure 12. The templates used in baseline (left) and the templates used in proposed method (right). Such change demonstrates the possibility of non-fixed aspect ratio templates, and they can be extended easily.

fixed aspect ratio templates (See **Figure 12**). It does not only change the appearance of output collage, but also fit the ROI content to the template as appropriate as it can be.

3.3.2 The Cost Function

Given a shot S and a template T, the cost function in [1] only measures the difference of size, that is, $C_{size} = |\text{Size}(T - S)|$. Where $Size$ is the "importance level" we have mentioned in **Table 3**. However, it can be replaced by any measure of difference between the target shot and the available template.

Our templates not only have various sizes, but also have various aspect ratios, to fit the shot into templates which have different aspect ratio, the shot ROI is first scaled along the short dimension, and then the ROI region must be extended along the other dimension to prevent distortion. Since we include those regions outside ROI, the unwanted areas will be counted (in pixels) into the cost. Given the scaled region S', the

cost function is then modified as:

$$C = \alpha * C_{size} + \beta * \text{Area}(T - S') \quad (7)$$

Where α and β are predefined weights and they are fixed.

3.3.3 Inter-Row Optimization vs. Intra-Row Optimization

The baseline approach can produce sufficient/diverse layout combinations on the media whose size (i.e. screen width) is large enough; however, for those mobile devices that have limited screen size, the generated solution (i.e. template combinations) usually lacks variety due to the limited solution space. Therefore, we introduce the inter-row optimization step into the original intra-row optimization.

Our idea is to punish the repeated row sequence in the minimization step, if a row sequence appeared twice, its cost will be multiplied by a coefficient σ, and so on. The minimization criterion is then modified by:

$$l = \arg\min_i \left[\left(\frac{1}{n_i} \sum_{j=1}^{n_i} c\left(f_{s+j-1}, q_{ij}\right) + w_i \right) * \sigma^{N-1} \right] \quad (8)$$

Where N is the number of times that a certain solution has appeared continuously. If a solution (i.e. template combination in a row) repeated many times, the algorithm above will tend to use another combination of templates, thus preventing the result collage from having a monotone layout.

3.4 Experiments

We collect a total of 32 videos (20 of them are talk videos in TED, 12 of them are popular movie trailers) for the following experiments. The talk dataset and the movie dataset have 156 shots and 42 shots in average, respectively.

The talk videos are suitable for summarization on mobile because their duration is usually longer and thus needs random access to recover the watching process if it is interrupted. Additionally, talk videos usually have a clear subject (e.g., speaker, pictures on the slide) so we can extract meaningful and effective ROIs from them. We also include movie trailers that are much more challenging into our experiments in order to evaluate a general situation. Some example shots can be referred in **Figure 4**.

3.4.1 Quantitative Evaluation

We expect that the proposed method can represent more informative contents while the space consumption remains near to the baseline. Several measurements have been proposed to evaluate our result. First, "ROI Ratio" is defined by:

$$ROI\ Ratio = \frac{1}{V}\sum_{i=1}^{V}\frac{1}{F}\sum_{j=1}^{F}\frac{Area_j^{ROI}}{Area_j} \quad (9)$$

Where F is the total number of frames in video i, and V is the total number of videos. Similarly, "Adapted ratio" is given by:

Table 4. Compactness Measurements Result. The first row represents the result of talk videos, and the second row is for the movie trailers.

Dataset	ROI ratio	Adapted ratio	Enlarged ratio	Collage area ratio
Talk	36%	52%	1.81	109%
Movie	38%	57%	1.70	104%

$$Adapted\ Ratio = \frac{1}{V}\sum_{i=1}^{V}\frac{1}{F}\sum_{j=1}^{F}\frac{Area_j^{Adapted}}{Area_j} \tag{10}$$

And "Enlarged ratio" is given by:

$$Enlarged\ Ratio = \frac{1}{V}\sum_{i=1}^{V}\frac{1}{F}\sum_{j=1}^{F}Scale_j \tag{11}$$

$Scale_j$ is the adjusted scale after adaptation of shot image j. Finally "Collage size ratio" is given by:

$$Collage\ Size\ Ratio = \frac{1}{V}\sum_{i=1}^{V}\frac{Collage\ Area_i^{Proposed}}{Collage\ Area_i^{Baseline}} \tag{12}$$

$Collage\ Area_i^{Collage}$ and $Collage\ Area_i^{Baseline}$ are the output collage size generated by proposed method and baseline, respectively. The quantitative results are shown in **Table 4**.

For column 2 and column 3, statistics show that after ROI extraction, more than 60% of the area is cropped out. However, the ROI cannot be directly put into the collage without adaptation due to the aspect ratio. After the adapt step, nearly half of the space in both datasets has been saved.

As for the last two columns, it shows that the content in the proposed method can give more clear subjects in the collage than the baseline while using the same space.

3.4.2 User Study

The usability of a summarization system (especially on mobile) is relatively subjective, so we also conduct a user study that includes several aspects to evaluate the proposed method.

We have invited 24 people: 15 of them are male, 9 of them are female. Their occupation distribution is: 6 undergraduates, 12 graduates, 4PhD, and 2 administrative stuffs.

We provide two identical smart phones to conduct this user study: HTC desire with Android 2.2 platform whose resolution is 800*480.

Figure 13. Illustration of user study. We provide 2 identical smart phones for users (left: baseline, right: proposed).

Four questions are listed below:

Q1. *Clearness* of both approaches.

Q2. The *Context Loss* in our approach.

Q3. The impression of templates with non-fixed aspect ratios.

Q4. The overall rating.

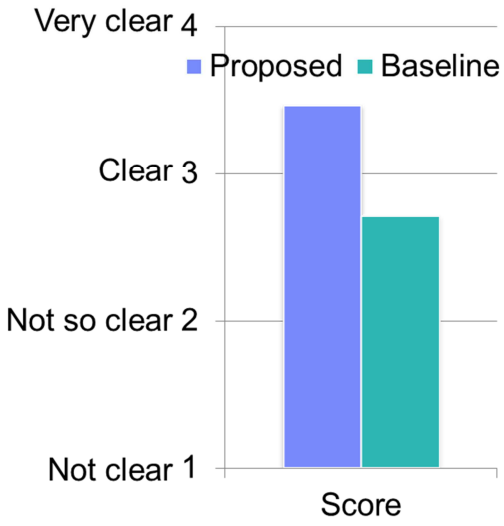


Figure 14. Q1 - The comparison of clearness in bar chart.

The first question asks user to evaluate the degree of clearness of the subject in the content, from 1 (not clear) to 4 (very clear). **Figure 14** shows the average score among 24 users. The baseline got a score near the borderline, while our approach was scored between “Clear” and “Very clear”.

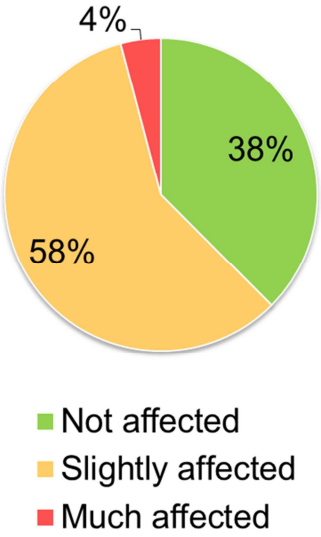


Figure 15. The result of Q2 in pie chart.

The second question is about the loss of context information. Although the proposed method can enlarge the content, it also makes the context cropped, so we are curious about how serious it is. The result (see **Figure 15**) shows that over half of users think that the context information of proposed method has been affected slightly by cropping ROI, nearly 40% people think that it is not affected, and only 4% (i.e. one person) think that it is seriously affected. Note that the cropping process is harmful for context information in general. However, the effect is not noticeable when such an application is in some environment with a limited space. In comparison with baseline, even though it keeps all context information, it is usually too small to be recognized. Only in some cases (e.g., a big scene that can distinguish the position of the subject) the baseline can maintain enough context information.

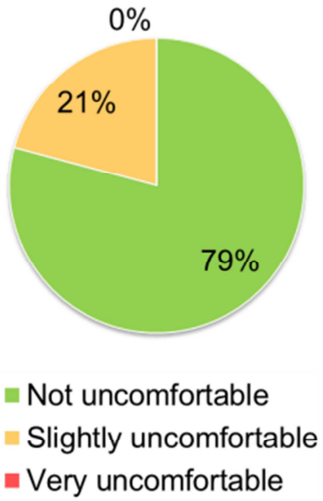


Figure 16. The result of Q3 in pie chart.

The third question is “Does changing aspect ratio affect your impression or does this arrangement make you uncomfortable?” We propose this question for we are concerned that users may want to stick with the original aspect ratio because they feel that all shots which have the fixed aspect ratio is much more like a video. Yet the result (see **Figure 16**) shows that nearly 80% people do not care about this issue.

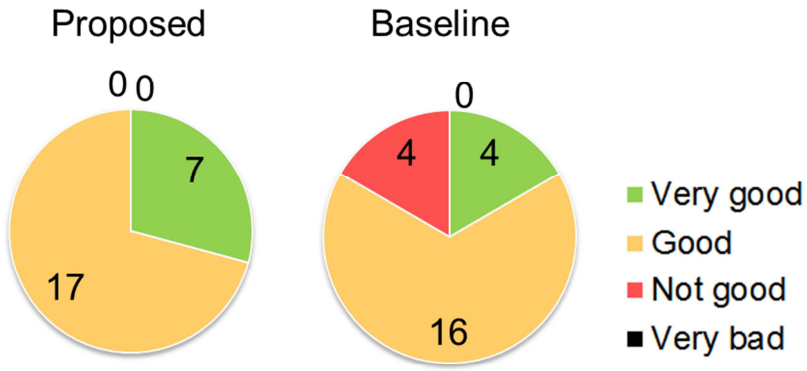


Figure 17. Q4 – Overall rating of both methods in pie chart.

The last question asks users to give an overall score for both methods. Although our method gets more "Very good" than baseline (7:4) and also has fewer negative scores (0:4), there are two-thirds of people that think they are both good (See **Figure 17**).

We have concluded some causes from users' feedback: The movie trailers are more attractive than talk videos, but the ROI extraction cannot give a satisfactory result in many complicated scenes that are mostly from movies. On the other hand, although we can extract effective ROIs from talk videos, they usually have monotone scenes (e.g., a speaker stands in front of a simple background), so the extracted ROI regions are likely to lose the diversity of content (e.g., most of the frames are the face of the speaker). Moreover, the face of the speaker is cropped in some cases. That is why our method does not significantly outperform the baseline in overall rating.

From users' feedback, we think that both of the two cases mainly result from the ROI extraction step. The ROI extraction tool which we used is for general purpose and

does not have any adjustment. Thus it can be further improved for the purpose of video summarization (e.g., applying face detection, extracting ROI from consecutive frames to make the ROI more robust, and so on).

Chapter 4

Conclusions and Future Work

Snap2Read demonstrates a possibility that people can turn the physical magazines into mobile e-book automatically and read them everywhere by simply snapping a shot. Compared to the text only e-books, our method can preserve layout appearance and images, free from being restricted by certain formats and hardware.

It is also possible to do magazine retrieval if there is a magazine database. Thus, if users see an interesting magazine by chance, they can retrieve parts of the magazine instead of buying them at full price. Furthermore, if we can apply image rectification techniques, the angle of inclination resulting from taking a snapshot will not be under so many restrictions as before through the help of mobile sensors, and the retrieval performance can also be improved.

We are developing the system for leveraging mobile sensors for boosting snapshot and rectification quality. Meanwhile, we are also evaluating the proposed mobile

reading system on Android phones for subjective performance.

Comp2Watch proposes a way to treat the video summarization on mobile environment which has limited space. ROI extraction is introduced to make it possible to place the shots on the tiny templates, and several key changes have been proposed to incorporate with the ROIs, thus improving the experience of watching videos on mobile devices.

Both the quantitative measure and the user study show that our method has a more clear result while using nearly the same space. The user study also shows that cropping out background (non-ROI regions) will not affect the understandability much.

The future works may include: Improve ROI extraction for our purpose as it mentioned in the last section, introduce image retargeting to be compared with cropping, and make the UI much more friendly (e.g., providing transcript if any, making the number of shot in a row manually adjustable). We think that these will make our work more robust and reliable.

Bibliography

[1] B. Erol, E. Antúnez and J. J. Hull, “HOTPAPER: Multimedia Interaction with Paper using Mobile Phones”, ACM Conference, 2008.

[2] C. Liao and Q. Liu, “PACER: Toward A Cameraphone-based Paper Interface for Fine-grained and Flexible Interaction with Documents”, ACM MM, 2009.

[3] X. Xie, G. Miao, R. Song, Ji-Rong Wen and Wei-Ying Ma, “Efficient Browsing of Web Search Results on Mobile Devices Based on Block Importance Model,” Proc. Pervasive Computing and Communications, IEEE, 2005.

[4] G. Hattori, K. Hoashi, K. Matsumoto, F. Sugaya, “Robust Web Page Segmentation for Mobile Terminal Using Content-Distances and Page Layout Information”, ACM WWW, 2007.

[5] O. Okun, D. Doermann and M. Pietikäinen, “Page Segmentation and Zone Classification: The State of the Art,” in UMD, 1999.

[6] Chih-Chung Chang and Chih-Jen Lin, “LIBSVM: a library for support vector machines”, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[7] A. Bosch, A. Zisserman and X. Munoz, “Representing shape with a spatial pyramid kernel”, CIVR, 2007.

[8] GEDI: Groundtruthing Editor http://gedigroundtruth.sourceforge.net/

[9] A. Antonacopoulos, B. Gatos and D. Bridson, “ICDAR2005 Page Segmentation Competition”, ICDAR, 2005.

[10] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky. Video Manga: generating semantically meaningful video summaries. In Proc. ACM Multimedia (MM), 1999. DOI= http://dx.doi.org/10.1145/319463.319654

[11] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake. AutoCollage. In Proc. ACM SIGGRAPH 2006. DOI= http://dx.doi.org/10.1145/1179352.1141965

[12] M. H. Lee, N. Singhal, S. Cho, and In Kyu Park. Mobile Photo Collage. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010. DOI= http://dx.doi.org/10.1109/CVPRW.2010.5543752

[13] S. Goferman, A. Tal, and L. Zelnik-Manor. Puzzle-like Collage. In EUROGRAPHICS, 2010.

[14] J. Harel, C. Koch, and P. Perona. Graph-Based Visual Saliency. In Proc. Neural Information Processing Systems (NIPS), 2006.

[15] S.C. Lee and S. Zhai. The Performance of Touch Screen Soft Buttons. In Proc. ACM Conference on Human Factors in Computing Systems (CHI), pages 309–318, 2009. DOI= http://dx.doi.org/10.1145/1518701.1518750

[16] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In Proc. ACM international conference on image and video retrieval (CIVR), 2007. DOI= http://dx.doi.org/10.1145/1282280.1282340

[17] Nielsen Company. State of the Media - Mobile Usage Trends: Q3 and Q4 2010. http://blog.nielsen.com/nielsenwire/online_mobile/number-of-americans-watching-mobile-video-grows-more-than-40-in-last-year/